Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack

Lab Objectives:

The main objective of this lab is to study the operation of the transport layer in the TCP/IP stack through:

1. Highlighting the concept of ports.
2. Generating a TCP connection using a web browser and observing the TCP protocol's three-way handshake.
3. Generating and analyzing a UDP connection.

Work Approach:

- A theoretical study covering general concepts and definitions of basic notions.
- A practical study to demonstrate the operational principles of the transport layer through the concept of ports and the two protocols, TCP and UDP.

I. Theoretical Part

I.1 The Transport Layer:

In a computer network, applications often use the simplified 5-layer TCP/IP model because the OSI model corresponds to a more theoretical approach that was developed earlier in the history of networking.

The transport layer ensures end-to-end communication between applications on machines connected within a computer network. This layer splits the data transmitted by the session layer (OSI model) or the application layer (TCP/IP stack) into smaller entities or segments and ensures that the elements are correctly delivered to the other side.

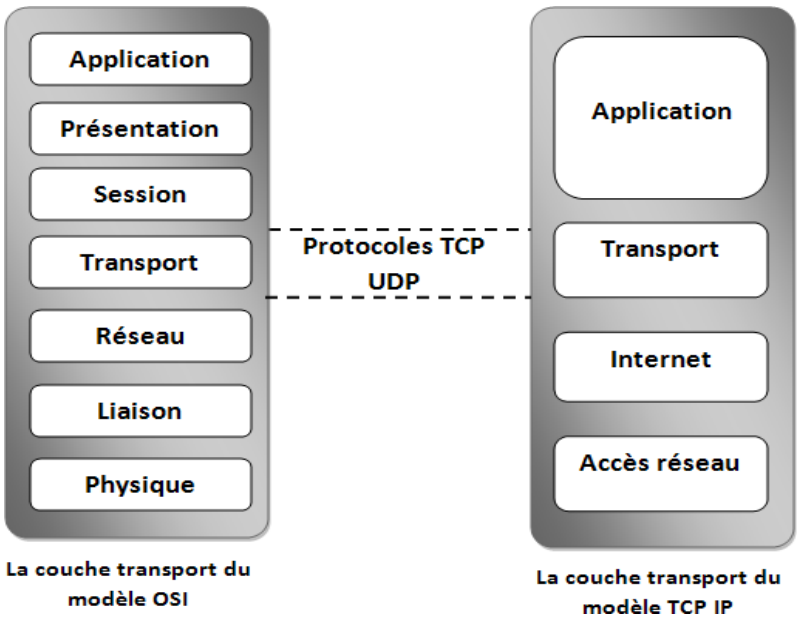


Figure 01: Overview of the Transport Layer

Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack

I.2 Transport Layer Protocols:

The transport layer includes two protocols that allow two applications to exchange data regardless of the type of network used (i.e., independently of the lower layers). These protocols are:

- The Transmission Control Protocol (TCP)
- The User Datagram Protocol (UDP)

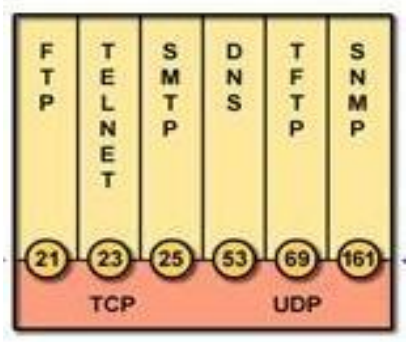
TCP is a reliable, connection-oriented protocol (connected mode) that ensures error-free delivery of packets from one machine in a network to another machine in the same network. Its role is to fragment the message to be transmitted in a way that allows it to pass through the internet layer. Conversely, on the destination machine, TCP reassembles the fragments received from the internet layer in the correct order to reconstruct the original message. TCP also handles the flow control of the connection.

UDP is a simpler protocol than TCP: it is unreliable and connectionless (connectionless mode). Its use assumes that there is no need for flow control or packet order preservation. For example, it is used when the application layer handles the reordering of messages. More generally, UDP is used when packet delivery time is a priority.

I.3 Concept of Ports:

A port is a unique address assigned to a specific application on a machine; this address is coded on 16 bits.

- Ports from 0 to 1023 are well-known or reserved ports. They are assigned by IANA (Internet Assigned Numbers Authority) and provide access to standard services (e.g., email via SMTP on port 25, web server via HTTP on port 80).
- Ports above 1024 are "user" ports available for running any application service.



Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack

II. Practical Part

Set up a client/server architecture between the machines in the lab to begin the following parts of this lab session.

II.1 Port Numbers:

In Linux, the /etc directory provides the configuration files necessary for system administration. For this lab, we will use subdirectories of this folder to identify a given application.

1. To find out the reserved port numbers for the most well-known protocols, simply view the /etc/services file (note that applications are available either in TCP or UDP).
2. Access the /etc/protocols file in a UNIX shell to retrieve the list of all protocols (our focus is on TCP and UDP protocols).

II.2 Capturing Frames with Wireshark:

As seen in the previous lab, Ethereal (now Wireshark) allows you to capture frames circulating on the network. In this part of the lab, we will use this software to study TCP segments and UDP messages from the transport layer.

II.2.1 Capturing TCP Packets:

1. **Wireshark Setup:**

a. Launch Wireshark using the command wireshark & (in the Linux shell in background mode) or launch Wireshark in Windows.
b. Select the interface to use for packet capture via the menu: Capture → Interfaces.
c. Click the Start button for the selected interface to begin capturing frames on the ETHERNET network.

2. **Establishing a Connection with the Server:**

a. In a browser, access the page 10.1.1.254 and the client/server applications from previous labs. (The command firefox & launches the browser in the background; then minimize the browser window and return to Wireshark if using Linux.)
b. The capture windows are now active. In Wireshark, locate the Source, Destination, and Protocol columns. The HTTP data carries the text and graphics of the page generated by the server, using TCP's reliability function.
c. In the Wireshark Capture menu, click Stop to end the capture.

3. **Generating Captured Packets:**

a. In this part, we will illustrate the TCP three-way handshake using the segment's flags (SEQ, SYN, and ACK). First, we need to isolate only the TCP packets using Wireshark's filter utility:

Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack

- To use a preconfigured filter, click the Analyze option in the menu, then click Display Filters.
- In the Display Filter window, click TCP only, then OK.

b. After filtering the TCP packets, scroll through the Wireshark window to the first captured TCP packet. This is the initial packet of the flow.

c. In the Info column, you will find the first TCP packet [SYN] from the source computer. The second packet is the server's response [SYN, ACK]. The third packet is the [ACK] from the source computer, completing the three-way handshake. Explain the meaning of these messages.

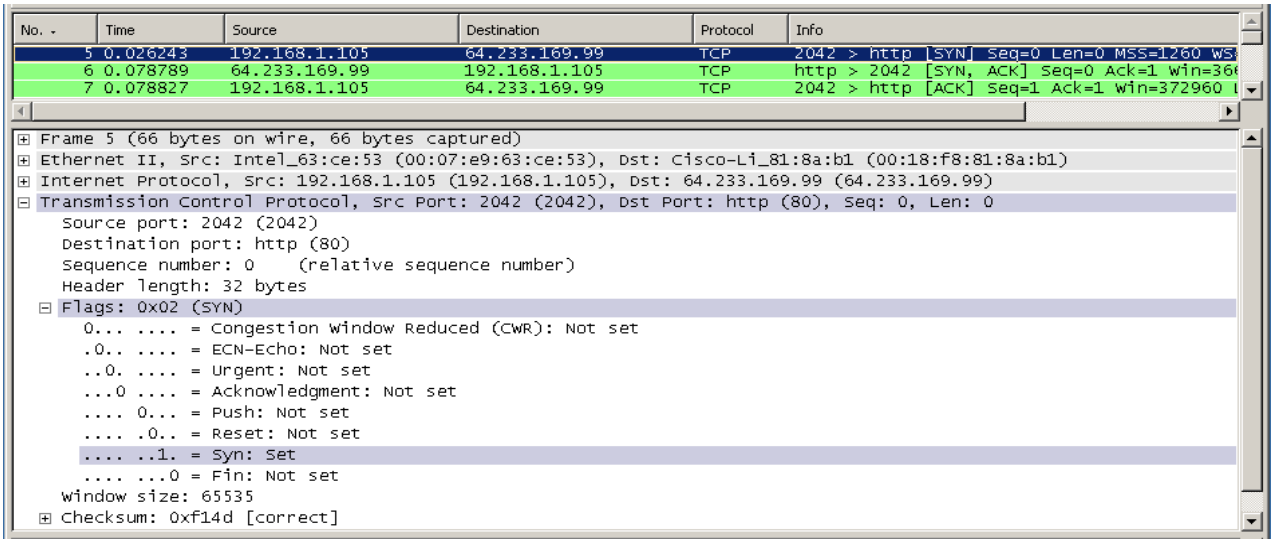
4. **Analyzing Captured Packets:**

To analyze the captured packets, use Wireshark's lower pane, which decodes information from the upper pane.

a. Click the + icon to expand the TCP (Transmission Control Protocol) details. Nearby, you will find the source and destination port numbers, sequence number (SEQ), and the length of the selected packet (len).

b. **Analysis of the First TCP Packet**: In the upper pane of Wireshark, select the first [SYN] packet of the TCP flow.

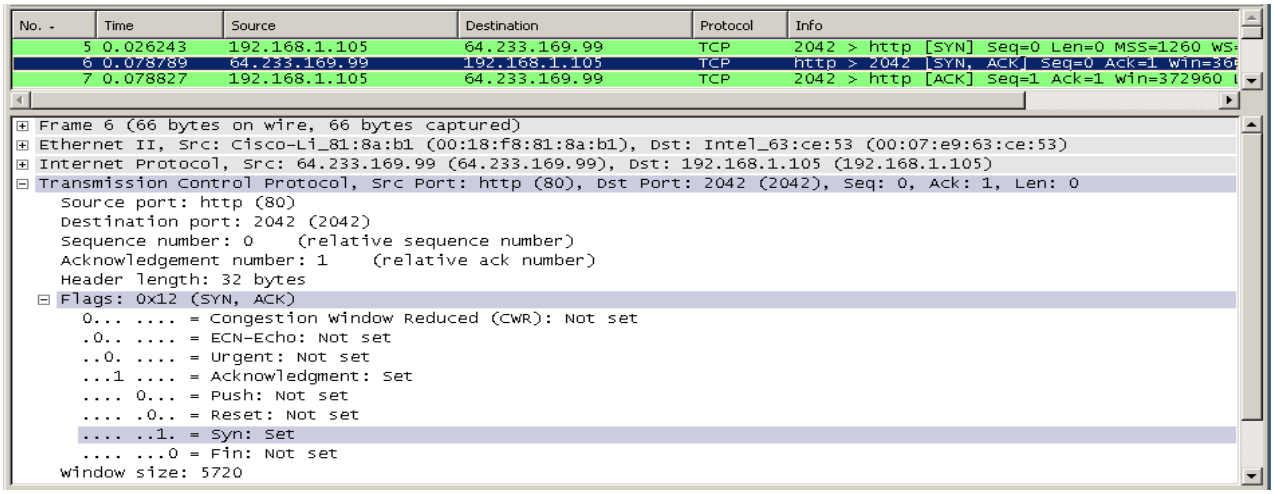
- Note that in the first TCP packet, the relative sequence number is set to 0.
- Click the + icon to expand the flags section and note that the SYN bit is set to 1.



c. **Analysis of the Second TCP Packet:** In the upper pane of Wireshark, select the second [SYN, ACK] packet of the TCP flow.

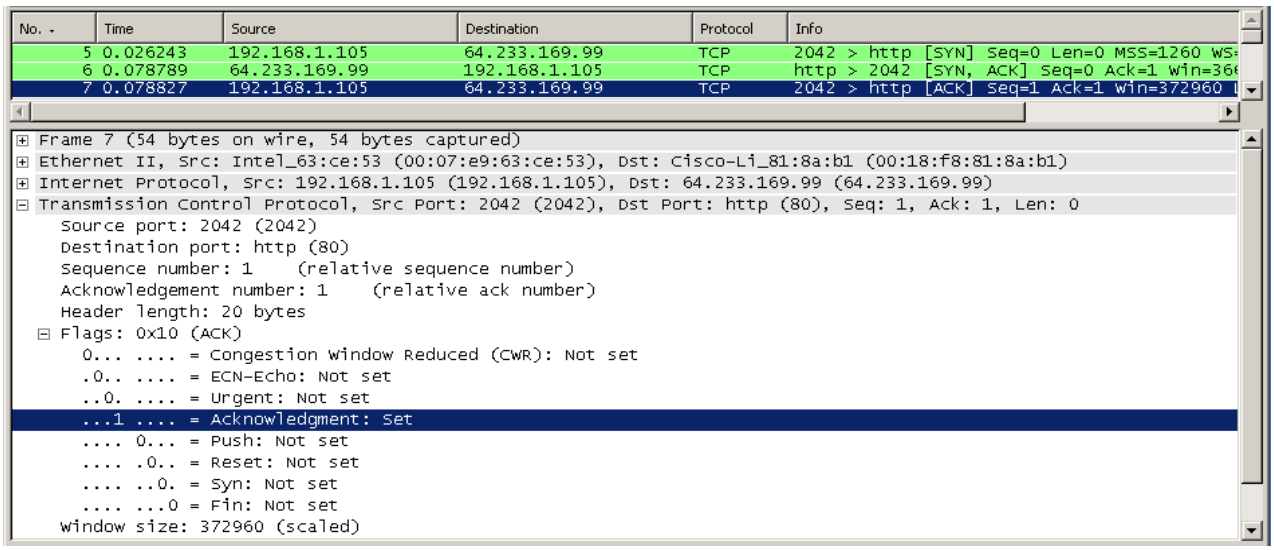
- Note that in this second packet of the three-way handshake, the relative sequence number is set to 0, and both the SYN and ACK bits are set to 1 in the flags field.

Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack



d. **Analysis of the Third TCP Packet:** In the upper pane of Wireshark, select the third [ACK] packet of the TCP flow.

- In this final packet of the handshake, the ACK (Acknowledgment) bit is set to 1, and the sequence number is set to 1, indicating that the TCP connection is now established and communication between the source computer and the server can begin.



Lab No. 7: Operation of the Transport Layer in the TCP/IP Stack

Exercise 1:

Fill in the following table based on your Wireshark results:

TCP Frame	Source Port	Destination Port	Flags	Value

II.2.2 Capturing UDP Packets:

1. Start a new capture for UDP frames only by using the UDP only filter under the menu Capture → Options as follows:
2. Open a console (or Command Prompt in Windows) and type a command like traceroute <IP address>. The choice of IP address is entirely up to you.
3. Stop the capture when the command prompt reappears in the console.
4. Save the capture file.

Exercise 02 (to be submitted in the report):

1. How many bytes are present in a UDP message?
2. Analyze the UDP packet and define its different fields.
3. Explain the role of the traceroute command by analyzing the UDP message.

Appendices:

1. Study of IPscan
2. Study of Psscan